

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号
特開2001-313634
(P2001-313634A)

(43) 公開日 平成13年11月9日 (2001.11.9)

(51) Int.Cl. ⁷	識別記号	F I	テーマコード* (参考)
H 0 4 L 9/08		G 0 6 F 15/00	3 3 0 E
G 0 6 F 15/00	3 3 0	H 0 4 L 9/00	6 0 1 C
H 0 4 L 9/32			6 7 3 A

審査請求 未請求 請求項の数33 O L (全 16 頁)

(21) 出願番号 特願2001-71600(P2001-71600)

(22) 出願日 平成13年3月14日 (2001.3.14)

(31) 優先権主張番号 60/190318

(32) 優先日 平成12年3月17日 (2000.3.17)

(33) 優先権主張国 米国 (U S)

(31) 優先権主張番号 09/638320

(32) 優先日 平成12年8月14日 (2000.8.14)

(33) 優先権主張国 米国 (U S)

(71) 出願人 596077259

ルーセント テクノロジーズ インコーポ
レイテッド
Lucent Technologies
Inc.

アメリカ合衆国 07974 ニュージャージ
ー、マレーヒル、マウンテン アベニュー
600-700

(74) 代理人 100081053

弁理士 三俣 弘文

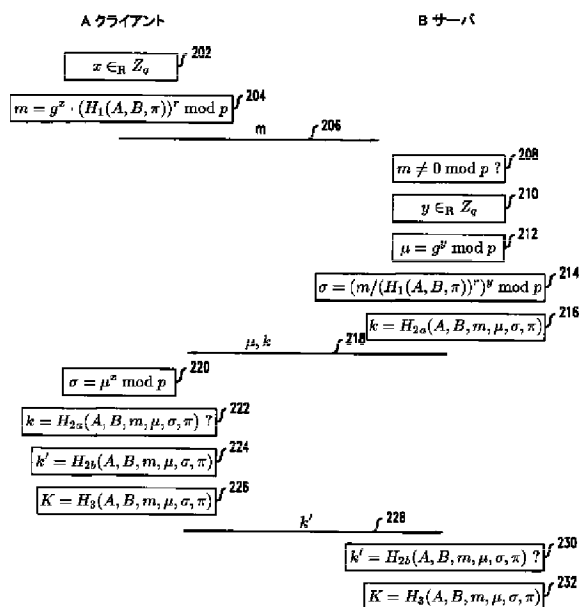
最終頁に続く

(54) 【発明の名称】 通信方法

(57) 【要約】

【課題】 安全であることが証明可能な、安全な、パスワードのみによる相互ネットワーク認証プロトコルを提供する。

【解決手段】 データネットワークを通じて、パスワードを共有するA、B二当事者間で、特定の群に関するDiffie-Hellman型鍵交換を用いて共有秘密 g^{xy} を生成する。g は両当事者に既知の群生成元であり、x は一方当事者A (例えばクライアント) に既知の指数であり、y は他方当事者B (例えばサーバ) に既知の指数である。Aが、 g^x と、少なくともパスワードの関数 H_1 に対する群演算を実行することによりパラメータmを生成し、mをBに送信する。これにより、Bは、mと、関数 H_1 に対して群逆演算を実行して g^x を抽出し、さらに、共有秘密 g^{xy} を計算することが可能となる。Aも、Bから g^y を受信し、共有秘密 g^{xy} を生成することができる。



【特許請求の範囲】

【請求項 1】 データネットワークを通じて、パスワードを共有する二当事者間で、特定の群に関するDiffie-Hellman型鍵交換を用いて共有秘密 g^{xy} を生成する通信方法において、

g は両当事者に既知の群生成元であり、 x は一方当事者に既知の指数であり、 y は他方当事者に既知の指数であり、前記群は群演算および群逆演算を有し、一方当事者が、 g^x と、少なくとも前記パスワードの関数とに対する群演算を実行することによりパラメータ m を生成し、 m を他方当事者に送信するステップを有し、これにより、他方当事者は、 m と、少なくとも前記パスワードの前記関数とに対して群逆演算を実行して g^x を抽出し、さらに、前記共有秘密 g^{xy} を計算することが可能であることを特徴とする通信方法。

【請求項 2】 前記一方当事者はクライアントであり、前記他方当事者はサーバであることを特徴とする請求項 1 記載の方法。

【請求項 3】 前記一方当事者が、前記他方当事者から g^y を受信し、前記共有秘密 g^{xy} を生成するステップをさらに有することを特徴とする請求項 1 記載の方法。

【請求項 4】 前記一方当事者が、受信値を、前記一方当事者の識別子、前記他方当事者の識別子、 m 、 g^y 、前記共有秘密、および前記パスワードのうちの少なくとも 1 つの関数と比較することによって、前記他方当事者を認証するステップをさらに有することを特徴とする請求項 3 記載の方法。

【請求項 5】 前記一方当事者が、前記一方当事者の識別子、前記他方当事者の識別子、 m 、 g^y 、前記共有秘密、および前記パスワードのうちの少なくとも 1 つの関数を、前記他方当事者に送信するステップをさらに有し、これにより、前記他方当事者は、前記一方当事者を認証することが可能であることを特徴とする請求項 3 記載の方法。

【請求項 6】 前記一方当事者が、前記一方当事者の識別子、前記他方当事者の識別子、 m 、 g^y 、前記共有秘密、および前記パスワードのうちの少なくとも 1 つの関数としてセッション鍵を生成するステップをさらに有することを特徴とする請求項 3 記載の方法。

【請求項 7】 g^y と、少なくとも前記パスワードの関数とに対して群演算を実行することにより前記他方当事者が計算したパラメータ μ を、前記一方当事者が受信するステップと、

前記一方当事者が、 μ と、少なくとも前記パスワードの前記関数とに対して群逆演算を実行して g^y を抽出するステップと、

前記一方当事者が、前記共有秘密 g^{xy} を計算するステップとをさらに有することを特徴とする請求項 1 記載の方法。

【請求項 8】 前記一方当事者が、前記一方当事者の識

別子、前記他方当事者の識別子、 m 、 μ 、前記共有秘密、および前記パスワードのうちの少なくとも 1 つの関数としてセッション鍵を生成するステップをさらに有することを特徴とする請求項 7 記載の方法。

【請求項 9】 データネットワークを通じて、二当事者間で、特定の群に関するDiffie-Hellman型鍵交換を用いて共有秘密 g^{xy} を生成する通信方法において、

g は両当事者に既知の群生成元であり、 x は一方当事者に既知の指数であり、 y は他方当事者に既知の指数であり、前記群は群演算および群逆演算を有し、

一方当事者が、 g^x と、少なくともパスワードベリファイアの関数とに対する群演算を実行することによりパラメータ m を生成し、 m を他方当事者に送信するステップを有し、これにより、他方当事者は、 m と、少なくとも前記パスワードベリファイアの前記関数とに対して群逆演算を実行して g^x を抽出し、さらに、前記共有秘密 g^{xy} を計算することが可能であることを特徴とする通信方法。

【請求項 10】 前記一方当事者はクライアントであり、前記他方当事者はサーバであることを特徴とする請求項 9 記載の方法。

【請求項 11】 前記一方当事者が、前記他方当事者から g^y を受信し、前記共有秘密 g^{xy} を生成するステップをさらに有することを特徴とする請求項 9 記載の方法。

【請求項 12】 前記一方当事者が、受信値を、前記一方当事者の識別子、前記他方当事者の識別子、 m 、 g^y 、前記共有秘密、および前記パスワードベリファイアのうちの少なくとも 1 つの関数と比較することによって、前記他方当事者を認証するステップをさらに有することを特徴とする請求項 11 記載の方法。

【請求項 13】 前記一方当事者に既知のインデックスを C として、前記一方当事者が、 g^C を計算するステップと、

前記一方当事者が、少なくとも g^C と、前記一方当事者の識別子、前記他方当事者の識別子、 m 、 g^y 、前記共有秘密、および前記パスワードベリファイアのうちの少なくとも 1 つの関数としてパラメータ e を計算するステップと、

前記パスワードベリファイアの離散対数を v として、前記一方当事者が、 $S = C - e \cdot v$ としてパラメータ S を計算するステップと、

前記一方当事者が、 S および g^C を前記他方当事者へ送信するステップとをさらに有し、これにより、前記他方当事者は、前記 S および g^C の値に少なくとも部分的に基づいて前記一方当事者を認証することが可能であることを特徴とする請求項 12 記載の方法。

【請求項 14】 前記一方当事者が、前記一方当事者の識別子、前記他方当事者の識別子、 m 、 g^y 、前記共有秘密、 g^C および前記パスワードベリファイアのうちの

少なくとも1つの関数としてセッション鍵を生成するステップをさらに有することを特徴とする請求項13記載の方法。

【請求項15】 前記一方当事者が、前記一方当事者の識別子、前記他方当事者の識別子、 m 、 g^y 、前記共有秘密、前記ランダム値、および前記パスワードベリファイアのうちの少なくとも1つの関数を、前記他方当事者に送信するステップをさらに有し、これにより、前記他方当事者は、前記一方当事者を認証することが可能であることを特徴とする請求項11記載の方法。

【請求項16】 前記パスワードベリファイアを公開鍵とし、前記パスワードベリファイアの離散対数を秘密鍵として用いてランダム値が暗号化される自己証明エルガマル暗号に基づいて、前記一方当事者は前記他方当事者を認証することを特徴とする請求項11記載の方法。

【請求項17】 前記一方当事者が、前記一方当事者の識別子、前記他方当事者の識別子、 m 、 g^y 、前記共有秘密、前記ランダム値、および前記パスワードベリファイアのうちの少なくとも1つの関数としてセッション鍵を生成するステップをさらに有することを特徴とする請求項16記載の方法。

【請求項18】 データネットワークを通じて、パスワードを共有する二当事者間で、特定の群に関するDiffie-Hellman型鍵交換を用いて共有秘密 g^{xy} を生成する通信方法において、 g は両当事者に既知の群生成元であり、 y は一方当事者に既知の指数であり、 x は他方当事者に既知の指数であり、前記群は群演算および群逆演算を有し、 g^x と、少なくとも前記パスワードの関数とに対して群演算を実行することにより前記他方当事者が計算したパラメータ m を、前記一方当事者が受信するステップと、前記一方当事者が、 m と、少なくとも前記パスワードの前記関数とに対して群逆演算を実行して g^x を抽出し、さらに、前記共有秘密 g^{xy} を計算するステップとを有することを特徴とする通信方法。

【請求項19】 前記一方当事者は前記パスワードを記憶することを特徴とする請求項18記載の方法。

【請求項20】 前記一方当事者は、少なくとも前記パスワードの前記関数を表す値を記憶するが、前記一方当事者は前記パスワードを記憶しないことを特徴とする請求項18記載の方法。

【請求項21】 前記一方当事者はサーバであり、前記他方当事者はクライアントであることを特徴とする請求項18記載の方法。

【請求項22】 前記一方当事者が、前記一方当事者の識別子、前記他方当事者の識別子、 m 、 g^y 、前記共有秘密、および前記パスワードのうちの少なくとも1つの関数を、前記他方当事者に送信するステップをさらに有することを特徴とする請求項18記載の方法。

【請求項23】 前記一方当事者が、受信値を、前記一方当事者の識別子、前記他方当事者の識別子、 m 、 g^y 、前記共有秘密、および前記パスワードのうちの少なくとも1つの関数と比較することによって、前記他方当事者を認証するステップをさらに有することを特徴とする請求項18記載の方法。

【請求項24】 前記一方当事者が、前記一方当事者の識別子、前記他方当事者の識別子、 m 、 g^y 、前記共有秘密、および前記パスワードのうちの少なくとも1つの関数としてセッション鍵を生成するステップをさらに有することを特徴とする請求項18記載の方法。

【請求項25】 前記一方当事者が、 g^y と、少なくとも前記パスワードの関数とに対する群演算を実行することによりパラメータ μ を生成し、 μ を前記他方当事者に送信するステップを有し、これにより、前記他方当事者は、 μ と、少なくとも前記パスワードの前記関数とに対して群逆演算を実行して g^y を抽出し、さらに、前記共有秘密 g^{xy} を計算することが可能であることを特徴とする請求項18記載の方法。

【請求項26】 データネットワークを通じて、二当事者間で、特定の群に関するDiffie-Hellman型鍵交換を用いて共有秘密 g^{xy} を生成する通信方法において、 g は両当事者に既知の群生成元であり、 y は一方当事者に既知の指数であり、 x は他方当事者に既知の指数であり、前記群は群演算および群逆演算を有し、 g^x と、少なくともパスワードベリファイアの関数とに対して群演算を実行することにより前記他方当事者が計算したパラメータ m を、前記一方当事者が受信するステップと、前記一方当事者が、 m と、少なくとも前記パスワードベリファイアの前記関数とに対して群逆演算を実行して g^x を抽出し、さらに、前記共有秘密 g^{xy} を計算するステップとを有することを特徴とする通信方法。

【請求項27】 前記一方当事者はサーバであり、前記他方当事者はクライアントであることを特徴とする請求項26記載の方法。

【請求項28】 前記一方当事者が、前記一方当事者の識別子、前記他方当事者の識別子、 m 、 g^y 、前記共有秘密、および前記パスワードベリファイアのうちの少なくとも1つの関数を、前記他方当事者に送信するステップをさらに有することを特徴とする請求項26記載の方法。

【請求項29】 C はランダム値であり、 v は、前記パスワードベリファイアの離散対数であり、 $S = C - e^v$ であり、 e は、少なくとも g^C と、前記一方当事者の識別子、前記他方当事者の識別子、 m 、 g^y 、前記共有秘密、および前記パスワードベリファイアのうちの少なくとも1つとの関数であるとして、前記一方当事者が、前記他方当事者から受信したパラメ

ータ S および g^c の値に少なくとも部分的に基づいて、前記他方当事者を認証するステップをさらに有することを特徴とする請求項26記載の方法。

【請求項30】 前記一方当事者が、前記一方当事者の識別子、前記他方当事者の識別子、 m 、 g^y 、前記共有秘密、 g^c および前記パスワードベリファイアのうちの少なくとも1つの関数としてセッション鍵を生成するステップをさらに有することを特徴とする請求項29記載の方法。

【請求項31】 前記一方当事者が、前記パスワードベリファイアを公開鍵とし、前記パスワードベリファイアの離散対数を秘密鍵として用いる自己証明エルガマル暗号を用いてランダム値を暗号化するステップと、暗号化されたランダム値を前記他方当事者へ送信するステップとをさらに有することを特徴とする請求項26記載の方法。

【請求項32】 前記一方当事者が、前記一方当事者の識別子、前記他方当事者の識別子、 m 、 g^y 、前記共有秘密、前記ランダム値、および前記パスワードベリファイアのうちの少なくとも1つの関数として前記他方当事者により計算された受信値に基づいて前記他方当事者を認証するステップをさらに有することを特徴とする請求項26記載の方法。

【請求項33】 前記一方当事者が、前記一方当事者の識別子、前記他方当事者の識別子、 m 、 g^y 、前記共有秘密、前記ランダム値、および前記パスワードベリファイアのうちの少なくとも1つの関数としてセッション鍵を生成するステップをさらに有することを特徴とする請求項26記載の方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、ネットワーク認証および鍵交換に関し、特に、パスワードのみによる安全な相互ネットワーク認証および鍵交換プロトコルに関する。

【0002】

【従来の技術】ネットワークを通じての認証は、リモートクライアントがネットワークサーバにアクセスすることを可能にするシステムのセキュリティ（安全性）の重要な部分である。認証は一般に、次のうちの1つまたは複数のものを確認することにより実行される。

- ・ユーザが知っているもの。例えば、パスワード。
- ・ユーザ自体。すなわち、指紋のような生体計測情報。
- ・ユーザが持っているもの。すなわち、スマートカードのような何らかの識別トークン。

例えば、現金自動預入支払機（ＡＴＭ）は、これらのうちの2つ、すなわち、ユーザが持っているもの（ＡＴＭカード）と、ユーザが知っているもの（個人識別番号（暗証番号、PIN））とを確認する。ＡＴＭ認証は、データネットワークを通じての認証よりもずっと容易で

ある。その理由は、ＡＴＭ自体は信頼されたハードウェアとみなされ、ＡＴＭカードの存在を確認し正しい情報を安全に中央取引サーバに転送することが信用されているからである。

【0003】認証に加えて、鍵交換は、データネットワークを通じての通信の重要な部分である。クライアントとサーバが認証された後、安全な通信チャネルがそれらの間に設定されなければならない。これは一般に、クライアントとサーバが、認証に続く通信の間に用いるためのセッション鍵と呼ばれるある鍵を交換することによって実行される。

【0004】データネットワーク、特に、インターネットのような公衆データネットワークを通じての認証は困難である。その理由は、クライアントとサーバ間の通信は、多くの異なるタイプの攻撃を受けやすいからである。例えば、盗聴(eavesdropping)攻撃では、敵は、クライアントとサーバ間の通信を傍受することにより、秘密情報を知る可能性がある。敵は、パスワード情報を知った場合、その情報をサーバに対して再生(リプレイ)して正当なクライアントになりすますことが可能となる。これをリプレイ攻撃という。リプレイ攻撃は、クライアントから送られたパスワードが暗号化されている場合でも有効である。その理由は、敵は、実際のパスワードを知る必要はなく、代わりに、サーバが正当なクライアントから期待しているもの（この場合、暗号化されたパスワード）をサーバに提供すればよいからである。もう1つのタイプの攻撃は、スプーフィング攻撃である。これは、敵がサーバになりすますことにより、クライアントは、正当なサーバと通信していると信じるが、実際のところは敵と通信しているというものである。このような攻撃では、クライアントは、要注意(sensitive)情報を敵に提供してしまう可能性がある。

【0005】さらに、パスワードに基づく認証プロトコルでは、パスワードが辞書攻撃を受けるくらい弱くなる可能性が存在する。辞書攻撃は、必要なパスワードに関する何らかの既知情報に対して、多数のありそうなパスワード（例えば、英語辞書中の全単語）をテストすることによって実行される、パスワードに対する力ずくの攻撃である。この既知情報は、公に利用可能なこともあるし、上記の技術のうちの1つにより敵が入手したものであることもある。ユーザはしばしば、覚えやすい、したがって推測しやすいパスワードを選択するので、辞書攻撃はしばしば有効である。

【0006】ネットワーク認証にはさまざまな技術が知られている。これらの公知技術は2つに分類される。第1の分類には、クライアントシステム上に永続的記憶データを要求する技術が含まれる。第2の分類には、クライアントシステム上に永続的記憶データを要求しない技術が含まれる。

【0007】第1の分類に関して、永続的記憶データに

は、決して漏れてはならない秘密(secret)データ（例えば、認証サーバと共有される秘密鍵）も、改竄を防止しなければならない、秘密ではないが要注意の(sensitive)データ（例えば、認証サーバの公開鍵）も含まれる。いずれのタイプの永続的データでも、データを敵からの攻撃から守るには、追加のセキュリティ条件が必要である。さらに、パスワードおよび永続的記憶データの両方に依拠する認証プロトコルを用いるときには、一方の漏洩が他方の弱点となることがある。例えば、秘密鍵が漏洩すると、パスワードに対する辞書攻撃が可能となることがある。この第1のクラスのプロトコルでのもう1つの問題点は、永続的記憶データは、鍵の生成および配送を必要とするが、これは面倒なことがあり、一般にシステムの柔軟性が低下することである。

【0008】第2の分類は、クライアントにおける永続的記憶データを要求しないため、パスワードのみによる認証プロトコルと呼ばれる。クライアントは、正当なパスワードを提供することができればよい。潜在的に弱いパスワードを用いて強力なセキュリティおよび認証を提供するということは、矛盾しているように思われる。しかし、安全であるように設計されたパスワードのみによるユーザ認証および鍵交換のプロトコルがいくつか存在する。これらのプロトコルは、D. Jablon, "Strong Password-Only Authenticated Key Exchange", ACM Computer Communication Review, ACM SIGCOMM, 26(5):5-20, 1996、に記載されている。これらのパスワードのみによるプロトコルのうち注目すべきものには次のものがある。

- ・ EKE (Encrypted Key Exchange) (S. M. Bellare and M. Merritt, "Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks", Proceedings of the IEEE Symposium on Research in Security and Privacy, pp. 72-84、に記載)

- ・ A-EKE (Augmented-EKE) (S. M. Bellare and M. Merritt, "Augmented Encrypted Key Exchange: A Password-Based Protocol Secure Against Dictionary Attacks and Password File Compromise", Proceedings of the First Annual Conference on Computer and Communications Security, 1993, pp. 244-250、に記載)

- ・ M-EKE (Modified EKE) (M. Steiner, G. Tsudik, and M. Waidner, "Refinement and Extension of Encrypted Key Exchange", ACM Operating System Review, 29:22-30, 1995、に記載)

- ・ SPEKE (Simple Password EKE) および DH-EKE (Diffie-Hellman EKE) (いずれも、D. Jablon, "Strong Password-Only Authenticated Key Exchange", ACM Computer Communication Review, ACM SIGCOMM, 26(5):5-20, 1996、に記載)

- ・ SRP (Secure Remote Password Protocol) (T. Wu, "The Secure Remote Password Protocol", Proceedings

s of the 1998 Internet Society Network and Distributed System Security Symposium, pp. 97-111, 1998、に記載)

- ・ OKE (Open Key Exchange) (Stefan Lucks, "Open Key Exchange: How to Defeat Dictionary Attacks Without Encrypting Public Keys", Security Protocol Workshop, Ecole Normale Supérieure, April 7-9, 1997、に記載)

【0009】

【発明が解決しようとする課題】これらの公知のパスワードのみによる認証プロトコルでの問題点は、これらは安全であることが証明されていないことである。実際、S. Patel, "Number Theoretic Attacks on Secure Password Schemes", Proceedings of the IEEE Symposium on Research in Security and Privacy, pp. 236-247, 1997、に記載されているように、EKEプロトコルは、いくつかの数論的攻撃を受ける可能性がある。ネットワークセキュリティの重要性の観点から、安全であることが証明可能な、パスワードのみによる相互認証プロトコルが必要である。

【0010】米国特許出願第09/353,468号（出願日：1999年7月13日）には、公開鍵暗号方式を利用した安全なパスワードのみによる相互ネットワーク認証プロトコルが記載されている。このプロトコルは、基礎となる公開鍵暗号方式と同程度に安全であることが証明されている。

【0011】

【課題を解決するための手段】本発明は、安全であることが証明可能な、安全な、パスワードのみによる相互ネットワーク認証プロトコルを提供する。本発明のプロトコルによれば、二当事者が、Diffie-Hellman型の鍵交換を用いて共有秘密を生成する。以下で詳細に説明するように、Diffie-Hellman型の鍵交換によれば、特定のいわゆる群に対する群生成元 g と、一方当事者に既知の指数 x と、他方当事者に既知の指数 y と、共有秘密 g^{xy} がある。一方当事者が g^x を生成し、他方当事者が g^y を生成し、両者はこれらの値を交換して、それぞれが共有秘密 g^{xy} を生成することができるようにする。Diffie-Hellmanは鍵交換プロトコルを定義しているが、このプロトコルは認証の観点を有しない。

【0012】本発明によれば、Diffie-Hellman型の共有秘密を用いるが、二当事者が共有パスワードを用いて互いを認証することができるように修正されたプロトコルが実現される。さらに、重要な点であるが、本発明の発明者は、このプロトコルが安全であることを証明した。本発明によれば、一方当事者がDiffie-Hellman値 g^x を生成し、これを、いわゆる群演算を用いて少なくともパスワードの関数と結合し、その結果得られる値を他方当事者へ送信する。群演算は、用いられる特定の群に対して定義され、以下で詳細に説明する。今の目的のために

は、あらゆる群は1つの群演算と、対応する群逆演算とを有することを認識すれば十分である。他方当事者は、値を受信すると、受信した値と、少なくともパスワードの関数とに対して群逆演算を実行して、 g^x を抽出する。これにより、他方当事者は、自己の y の知識を用いて、共有秘密 g^{xy} を生成することができる。ここに記載するように、群演算および群逆演算をDiffie-Hellman型の鍵交換プロトコルとともに用いることは、従来技術のパスワードのみによる相互ネットワーク認証プロトコルに比べて利点がある。おそらく最も重要な点は、通信チャンネルにアクセス可能な敵による攻撃に対して安全であることを証明することができるプロトコルを提供する点である。上記のように、Diffie-Hellman値 g^x は、少なくともパスワードの関数と結合される。「少なくとも」という用語を用いるのは、さまざまな実施例において、 g^x は、パスワードのみの関数と結合されることもあるが、パスワードが特定の当事者のペアに対して一意であることを保証するために、パスワードと、プロトコルに対する当事者の識別子との関数と結合されることもあるからである。

【0013】本発明の一実施例によれば、当事者は、少なくともいくつかのパラメータの関数を計算し、計算した値を他方当事者に送信し、受信された値を各当事者が自己の計算値に照らしてチェックすることによって、互いを認証することが可能である。この計算に用いられるパラメータは、少なくとも1つの当事者識別子、Diffie-Hellman値(g^x または g^y)、共有秘密、および共有パスワードとすることが可能である。これらの値のうちの少なくとも1つの関数を計算することにより、当事者は、他方当事者が共有パスワードを所有していることを認証することが可能である。

【0014】本発明のもう1つの実施例によれば、当事者は、明示的に互いを認証するのではなく、代わりに、当事者は、それぞれ共有秘密鍵を生成し、生成した共有秘密鍵を通信のためのセッション鍵として使用することによって、暗黙的に互いを認証する。いずれかの当事者が正しいパスワードを所有していない場合、その当事者は正しい秘密セッション鍵を生成することができず、当事者間の通信は不可能となる。この実施例によれば、両当事者が、群演算を用いて自己のDiffie-Hellman値を少なくともパスワードの関数と結合し、結果として得られる値を他方当事者へ送信するという上記の技術を使用する。他方当事者から値を受信すると、各当事者は、群逆演算を用いて他方当事者のDiffie-Hellman値を抽出し、共有秘密鍵を計算する。

【0015】通信プロトコルに対する二当事者は、ほとんどの場合、クライアントコンピュータとサーバコンピュータである。上記の実施例では、クライアントおよびサーバはいずれも共有パスワードを記憶する。本発明の他の実施例では、サーバにおけるセキュリティ劣化に対

する保護を行うため、サーバは、パスワードを所有せず、代わりに、いわゆるパスワードベリファイアを与えられ、記憶する。パスワードベリファイアは、以下で詳細に説明するように、パスワードのある関数である。パスワード自体は、パスワードベリファイアの知識から決定することはできない。本発明のこれらの実施例によるプロトコルは、パスワードベリファイアが一般に実際のパスワードの代わりに用いられることを除いては、上記の実施例と類似している。しかし、サーバが実際のパスワードを知らないため、他方当事者が実際に正しいパスワードベリファイアまたは実際のパスワードを所有していることを各当事者が安全に認証するためには、異なる技術が二当事者によって使用されなければならない。一実施例では、当事者は、エルガマル暗号技術に基づく暗号化を用いて互いを認証する。

【0016】

【発明の実施の形態】暗号方式は、二当事者間に安全な通信を提供するための周知の技術である。本発明のさまざまな実施例について説明する前に、いくつかの背景知識および基本的用語について説明する。

【0017】略式には、集合 S から集合 T への関数 f について、 S に属するすべての x に対しては $f(x)$ は計算が容易であるが、 T に属するほとんどの y に対しては、 $f(x) = y$ となるような S の元 y を見つけるのが計算量的に実行不能である場合、 f は一方方向性関数であるという。一方方向性関数の一例は、法指数演算である。 p を大きい素数とし、 g を、 p を法とする乗法群(すなわち、 $1, \dots, p-1$ の範囲の数)の生成元とする。この場合、 $f(x) = g^x \bmod p$ は一般に、一方方向性関数であると仮定される。その逆関数(離散対数関数と呼ばれる)は、計算が困難である。離散対数関数の計算が困難な他の群もある。例えば、ある種の楕円曲線群である。

【0018】 k および l をセキュリティパラメータとする。ただし、 k は、主セキュリティパラメータであり、ハッシュ関数および秘密鍵に対する一般的なセキュリティパラメータとみなされ、 $l > k$ は、離散対数に基づく公開鍵に対するセキュリティパラメータとみなされる。

$\{0, 1\}^*$ を、有限二進数列の集合とし、 $\{0, 1\}^n$ を、長さ n の二進数列の集合とする。実数値関数 $\epsilon(n)$ は、任意の $c > 0$ について、すべての $n > n_c$ に対して $\epsilon(n) < 1/n_c$ となるような $n_c > 0$ がある場合、無視しうるといふ。少なくともサイズ k の q と、サイズ l の p を、ある値 r に対して $p = r \cdot q + 1$ が q と互いに素であるような素数とする。 g を、 \mathbb{Z}_p^* の、サイズ q の部分群の生成元とする。この部分群を $G_{p,q}$ と呼ぶことにする。

【0019】Diffie-Hellman鍵交換と呼ばれる鍵交換プロトコル(W. Diffie and M. Hellman, "New Directions in Cryptography", IEEE Transactions on Informati

on Theory, vol. 22, no. 6, 644-654, 1976、に記載)
 は、法指数関数に基づいている。具体的には、二当事者
 AとBが、図1に記載するプロトコルに従って、秘密鍵
 について合意する。ステップ102で、Aは、群 Z_q か
 らランダムに x を選ぶ。ただし、 $Z_q = \{0, 1, \dots, q-1\}$ (あるいは単に、 q を法とする整
 数)である。ステップ104で、Aは、 $X = g^x \bmod p$
 を計算する。ステップ106で、Aは、 X をBへ
 送信する。ステップ108で、Bは、 Z_q からランダム
 に y を選ぶ。Bは、ステップ110で、 $Y = g^y \bmod p$
 を計算し、ステップ112で、 Y をAへ送信す
 る。この時点で、共有秘密 g^{xy} (すなわち、秘密鍵)
 が、AとBの両者により計算可能となる。(なお、以
 下、 $\bmod p$ で考えていることが明らかである場合に
 は、記法を簡単にするため、 $\bmod p$ の記載を省略す
 ることがある。) $X = g^x$ がステップ106でAからB
 へ送信されたため、ステップ116で、Bは、 X^y を計
 算することにより、共有秘密 g^{xy} を計算することがで
 きる。同様に、 $Y = g^y$ がステップ112でBからAへ
 送信されたため、ステップ114で、Aは、 Y^x を計算
 することにより、共有秘密 g^{xy} を計算することができ
 る。こうして、共有秘密 S は、安全な通信のためのセッ
 ション鍵として、AとBにより用いられることが可能と
 なる。

【0020】Diffie-Hellman鍵交換は、ある種の楕円曲
 線群のような、離散対数関数の計算が困難な他の群の上
 でも実行可能である。群は当業者に周知であり、I. N.
 Herstein, "Topics in Algebra", 2nd edition, John W
 iley and Sons, New York, 1975、には次のように記載
 されている。空でない元の集合 G において、積と呼ばれ
 ・で表される次のような二項演算が定義される場合、 G
 は群をなすという。

1. $a, b \in G$ ならば、 $a \cdot b \in G$ (閉じている)。
2. $a, b, c \in G$ ならば、 $a \cdot (b \cdot c) = (a \cdot b) \cdot c$ (結合律)
3. すべての $a \in G$ に対して、 $a \cdot e = e \cdot a = a$ とな
 るような元 e が存在する (G における単位元の存在)。
4. 任意の $a \in G$ に対して、 $a \cdot a^{-1} = a^{-1} \cdot a = e$
 となるような元 a^{-1} が存在する (G における逆元の
 存在)。

【0021】こうして、より一般的に、Diffie-Hellman
 鍵交換は、特定の群において作用する。ただし、秘密鍵
 x および y は、群の元の指数である。そこで、 $g \in G$ を
 群生成元とする群 $G = \{g, g \cdot g, g \cdot g \cdot g, g \cdot g \cdot g \cdot g, \dots\}$
 を考える。ここで、 \cdot は群演算である。例として、 G の群演算が乗算である場合、 $G = \{g^1, g^2, g^3, g^4, \dots\}$
 である。 G の群演算が加算である場合、 $G = \{1g, 2g, 3g, 4g, \dots\}$
 である。本発明は、さまざまな群を用いて実施可能であ
 るため、以下の説明および特許請求の範囲において、記

法 g^x は、群生成元 g に対して群演算を x 回作用させる
 ことを意味する。さらに、任意の群について、ここで/
 で表す群逆演算も存在する。以下の説明および特許請求
 の範囲において、群逆演算は次のように定義される。す
 なわち、 x および y に対する群逆演算、すなわち、 $x \backslash$
 y は、 $x \cdot y^{-1}$ として定義される。

【0022】本発明の第1実施例による相互認証および
 鍵交換プロトコルを図2に示す。図の左側に示すステッ
 プは第1当事者Aによって実行され、図の右側に示すステ
 ップは第2当事者Bによって実行される。通常、Aは
 クライアントマシンとなり、Bはサーバマシンとなる。
 しかし、これは必ずしも必要ではなく、AおよびBは、
 代表的な場合を示す例としてそれぞれクライアントおよ
 びサーバとラベルされているだけである。理解されるよ
 うに、本発明は、AおよびBがクライアントおよびサー
 バである場合に限定されるものではなく、任意の二当事
 者AおよびBに適用可能である。矢印は、当事者間の通
 信を表す。このプロトコルによれば、サーバは自分自身
 をクライアントに認証してもらい、クライアントは自分
 自身をサーバに認証してもらう。両者の認証が完了した
 後、それぞれが、後の安全な通信に使用可能な秘密セッ
 ション鍵を生成することになる。

【0023】プロトコルを開始する前に、クライアント
 およびサーバは、クライアントがサーバとの認証に用い
 るパスワード π を所有すると仮定する。

【0024】なお、以下のプロトコルは、サーバおよび
 クライアントの両方を認証する。したがって、サーバお
 よびクライアントのいずれも真正でないと仮定され、こ
 のため、サーバまたはクライアントのいずれかが敵であ
 る可能性がある。クライアントは、自分自身を認証して
 もらってサーバにアクセスしようとする敵である可能性
 がある。サーバは、疑いを持たないクライアントから要
 注意情報を得ようとして別の真正なサーバのふりをしよ
 うとする敵である可能性がある。

【0025】当業者には容易に明らかになるように、サ
 ーバおよびクライアントは、コンピュータプログラムコー
 ドの制御下で動作するプログラムされたコンピュータ
 として実装可能である。コンピュータプログラムコード
 は、コンピュータ可読媒体 (例えば、メモリ) に記憶さ
 れ、コードは、コンピュータのプロセッサにより実行さ
 れる。本発明の明細書の記載に基づいて、当業者であれ
 ば、ここで説明するプロトコルを実装するために適当な
 コンピュータプログラムコードを容易に作成することが
 可能である。クライアントとサーバは、データネットワ
 ークを通じて互いに通信する。このようにネットワーク
 接続された、プログラムされたコンピュータは当業者に
 周知であるため、ここでは詳細には説明しない。

【0026】図2に戻り、ステップ202で、クライア
 ントは、 Z_q から指数 x としてランダムな値を選ぶ。次
 に、ステップ204で、クライアントは、 $m = g^x \cdot$

$(H_1(A, B, \pi))^r \bmod p$ として、パラメータ m を計算する。ただし、 A はクライアントの一意的識別子であり、 B はサーバの一意的識別子であり、 π は、この特定のサーバに対するクライアントのパスワードであり、 H_1 はランダムハッシュ関数であり、 \cdot は群演算を表す。 $H_1(A, B, \pi)$ は、結果が G_{1p} 内にいることを保証するために、 r 乗される。略式には、集合 S から集合 T への関数 H について、 H の出力がランダムに見える場合、すなわち、少なくとも、 S に属する入力 x で関数が計算されるまでは予測不能である場合、ランダムハッシュ関数と呼ばれる。 H_1 は、 Z_p^* においてランダムに見えるものを出力しなければならないため、 $|p| + \text{sec}$ ビット（ただし、 $|p|$ は p のビット数であり、 sec はセキュリティパラメータである）を出力すべきである。セキュリティパラメータは、例えば、160とすることが可能である。一般にこのようにふるまう既知の関数には次のものがある。

- ・SHA-1 (FIPS 180-1, "Secure Hash Standard", Federal Information Processing Standards Publication 180-1, 1995、に記載)

- ・RIPEMD-160 (H. Dobbertin, A. Bosselaers, B. Preneel, "RIPEMD-160: a strengthened version of RIPEMD", Fast Software Encryption, 3rd Intl. Workshop, 71-82, 1996、に記載)

【0027】各クライアント・サーバ対ごとに一意的であることを保証するために、パスワードのみではなく、組(タプル) (A, B, π) を用いる。発見的セキュリティに要求されると思われるのはパスワードのみであるが、以下で詳細に説明するように、セキュリティの形式的証明には、クライアントおよびサーバの名前が必要であると思われる。したがって、本発明の1つの特徴によれば、少なくともパスワードの関数と、Diffie-Hellman値 g^x に対して群演算を実行することにより、少なくともパスワードの関数をDiffie-Hellman値 g^x と結合する。これは、パスワードの知識を有する者によってのみDiffie-Hellman値 g^x がパラメータ m から抽出可能であることを保証するので、プロトコルの重要なステップである。このDiffie-Hellman値 g^x の抽出については、ステップ214に関して以下で詳細に説明する。ステップ206で、クライアントは、パラメータ m をサーバへ送信する。

【0028】パラメータ m を受信すると、ステップ208で、サーバは、パラメータ値が $0 \bmod p$ でないことを保証するために、パラメータ値をテストする。この値が $0 \bmod p$ である場合、サーバはプロトコルを終了する。 0 は Z_p^* に属さないからである。この値が $0 \bmod p$ でない場合、ステップ210で、サーバは、 Z_q から指数 y としてランダムな値を選ぶ。ステップ212で、サーバは、パラメータ μ にDiffie-Hellman値 g^y を代入する。次に、ステップ214で、サーバ

は、受信したパラメータ m を用いて、次式のように、Diffie-Hellman共有秘密 g^{xy} （このプロトコルでは σ で表す）を計算する。

$$\sigma = (m / (H_1(A, B, \pi))^r)^y \bmod p$$

次に、このステップについて、さらに詳細に説明する（記法を簡単にするため $\bmod p$ の記載を省略する）。まず、想起されるべき点であるが、上記のように、任意の群演算に対して、 x および y に対する群逆演算、すなわち、 x/y が $x \cdot y^{-1}$ として定義されるような群逆演算がある。そこで、当業者には認識されるように、ステップ214における $m / (H_1(A, B, \pi))^r$ の計算は、 m と、少なくともパスワードの関数とに対する群逆演算を実行している。ステップ204からの m の値を代入すると、 $g^x \cdot (H_1(A, B, \pi))^r / (H_1(A, B, \pi))^r = g^x$ を得る。したがって、サーバは、正しいパスワード π を所有している場合、受信したパラメータ m の値からDiffie-Hellman値 g^x を抽出することができる。こうして、ステップ214における計算の結果、サーバは、Diffie-Hellman共有秘密 g^{xy} を生成する。

【0029】次に、ステップ216で、サーバは、 $k = H_{2a}(A, B, m, \mu, \sigma, \pi)$ を計算する。ただし、 H_{2a} は、 sec ビット（ただし sec はセキュリティパラメータ）を出力しなければならないもう1つのランダムハッシュ関数である。パラメータ k は、以下で説明するように、サーバが正しいパスワードを所有していることを認証するために、クライアントAにより使用されることになる。ステップ218で、サーバは、パラメータ μ および k をクライアントへ送信する。

【0030】パラメータ μ および k を受信すると、クライアントは、ステップ220で、 $\sigma = \mu^x \bmod p$ を計算する。 $\mu = g^y$ であるため、 $\mu^x = g^{xy}$ はDiffie-Hellman共有秘密である。ステップ222で、クライアントは、自己の π の知識を用いて $H_{2a}(A, B, m, \mu, \sigma, \pi)$ を計算し、その結果が、ステップ218でサーバから受信したパラメータ k に等しいかどうかをテストする。これらが等しい場合、クライアントはサーバを認証したことになる。これらが等しくない場合、サーバは認証されなかったため、クライアントはプロトコルを終了する。ステップ224で、クライアントは $k' = H_{2b}(A, B, m, \mu, \sigma, \pi)$ を計算する。これは、以下で説明するように、サーバがクライアントを認証するために用いられることになる。ステップ226で、クライアントは、 $K = H_3(A, B, m, \mu, \sigma, \pi)$ としてセッション鍵 K を生成する。ステップ228で、クライアントは、 k' をサーバへ送信する。今度も、 H_{2b} および H_3 は、 sec ビット（ただし sec はセキュリティパラメータ）を出力しなければならないランダムハッシュ関数である。

【0031】ステップ230で、サーバは、自己の π の

知識を用いて $H_{2b}(A, B, m, \mu, \sigma, \pi)$ を計算し、その結果が、ステップ 228 でサーバから受信したパラメータ k' に等しいかどうかをテストする。これらが等しい場合、サーバはクライアントを認証したことになる。これらが等しくない場合、クライアントは認証されなかったため、サーバはプロトコルを終了する。ステップ 232 で、サーバは、 $K = H_3(A, B, m, \mu, \sigma, \pi)$ としてセッション鍵 K を生成する。

【0032】この時点で、クライアントとサーバの両者は互いを認証したことになり、クライアントとサーバはいずれも同一の安全なセッション鍵 K を生成している。これは、クライアントとサーバの間の後の安全な通信に用いることが可能である。

【0033】次に、本発明の第2実施例について図3を参照して説明する。この実施例では、当事者間の認証は暗黙的である。これは、一方当事者が他方当事者を明示的に認証するステップがないことを意味する。むしろ、各当事者はセッション鍵を生成しようとするが、いずれかの当事者が正しいパスワードを有しない場合、その当事者は正しいセッション鍵を生成することができず、当事者間の通信が不可能となる。図3を参照すると、ステップ 302 ~ 310 は、図2のステップ 202 ~ 210 に対応する。ステップ 312 で、サーバは、 $\mu = g^y \cdot (H_2(A, B, \pi))^r \bmod p$ としてパラメータ μ を計算する。これは、図2に関して前述したステップ 204 と同様である。ステップ 204 に関して前述したように、実際には、ステップ 312 は、少なくともパスワードの関数と、Diffie-Hellman 値 g^y に対して群演算を実行することにより、少なくともパスワードの関数を Diffie-Hellman 値 g^y と結合することになる。

【0034】次に、ステップ 314 で、サーバは、ステップ 214 に関して前述したのと同様に、Diffie-Hellman 値 g^x を抽出し、Diffie-Hellman 共有秘密 g^{xy} を計算する。ステップ 316 で、サーバは、 $K = H_3(A, B, m, \mu, \sigma, \pi)$ としてセッション鍵 K を計算する。ステップ 318 で、サーバは、ステップ 312 で生成したパラメータ μ をクライアントへ送信する。

【0035】パラメータ μ を受信すると、ステップ 320 で、クライアントは、パラメータ μ の値が $0 \bmod p$ でないことを保証するために、このパラメータ値をテストする。この値が $0 \bmod p$ である場合、クライアントはプロトコルを終了する。この値が $0 \bmod p$ でない場合、ステップ 322 で、クライアントは、 μ の値を用いて、前述のように、Diffie-Hellman 値 g^y を抽出し、Diffie-Hellman 共有秘密 g^{xy} を計算する。ステップ 324 で、クライアントは、 $K = H_3(A, B, m, \mu, \sigma, \pi)$ としてセッション鍵 K を計算する。

【0036】この時点で、クライアントおよびサーバはいずれも、パスワードの知識に基づいて、セッション鍵

K を生成したことになる。クライアントおよびサーバが正しいパスワードを所有していたとすれば、両者は同じセッション鍵 K を生成していることになり、これを用いて安全な通信が可能である。しかし、当事者の一方が正しいパスワードを所有していなかった場合、その当事者は、正しいセッション鍵を生成しておらず、当事者間の通信が不可能となる。

【0037】図2および図3に関して説明したプロトコルは、サーバがパスワード π を所有し記憶していることを仮定している。このようなプロトコルの1つの潜在的な問題点は、サーバ記憶領域のセキュリティ劣化により、敵がクライアントのパスワードを取得する可能性があることである。このような事態に対する保護を行うため、次に説明する本発明のもう1つの実施例では、サーバはパスワード π を所有せず、代わりに、いわゆるパスワードベリファイア V を記憶する。ここで、パスワードベリファイアとは、パスワードの関数として計算可能な値であるが、パスワードをパスワードベリファイアの知識から決定することはできない。ここで説明する実施例では、パスワードベリファイア V は、パスワード π のある関数である。ただし、特定のクライアント A およびサーバ B に対して、 $V = g^v$ 、 $v = H_0(A, B, \pi)$ であり、 H_0 は、 $|q| + \text{sec}$ ビットを出力しなければならないランダムハッシュ関数である。クライアントは、 π を知っているため、 $v = H_0(A, B, \pi)$ を計算することができ、さらに、 v からベリファイア $V = g^v$ を計算することができる。サーバは V しか知らないため、 v （これは、 V の離散対数である）を計算することはできない。本発明のこの実施例によるプロトコルを図4に示す。まず、ステップ 401 で、クライアントは、上記のようにパスワードベリファイア V を生成する。ステップ 402 ~ 422 は、ステップ 202 ~ 222 に関して前述したように実行される。ただし、ステップ 402 ~ 422 では、パスワード π をパスワードベリファイア V で置き換える。この時点で、ステップ 422 でのテストが真である場合、クライアントは、サーバが正しいパスワードベリファイア V を知っていることを認証したことになり、次に、クライアントは、自分が正しいパスワードを知っていることを証明することによって、自分自身をサーバに認証してもらわなければならない。

【0038】ステップ 424 で、クライアントは、 Z から指数 C としてランダムな値を選ぶ。クライアントは、ステップ 426 で、 $a = g^C$ を計算し、ステップ 428 で、 $e = H(A, B, m, \mu, \sigma, a, V)$ を計算する。次に、ステップ 430 で、クライアントは、 $S = C - ev$ を計算する。ステップ 432 で、クライアントは、 S および a をサーバへ送信する。 S および a を受信すると、ステップ 434 で、サーバは、自己の V の知識を用いて、 $e = H(A, B, m, \mu, \sigma, a, V)$ を計算する。ステップ 436 で、サーバは、 g^{SV} を計算

する。ステップ436でサーバにより計算された値が、ステップ432でクライアントから受信したaの値と一致する場合、サーバは、クライアントを真正なものとして受け入れる。最後に、クライアントおよびサーバは、それぞれステップ438および440で、 $K=H_3(A, B, m, \mu, \sigma, a, V)$ としてセッション鍵を計算する。

【0039】直観的には、ステップ436でのテストは、クライアントを次のようにして認証することになる。ステップ436での $a=g^S V^v$ というサーバの計算を参照すると、 $V=g^v$ であるため、ステップ436での計算は、 $a=g^S (g^v)^v = g^S g^{vv} = g^{S+vv}$ となる。ステップ430から、 $C=S+e v$ であることがわかるからである。したがって、サー

バによるaの計算結果が、ステップ432でクライアントから受信したaと一致した場合、サーバは、クライアントがvの知識を有していることを知る。vは、クライアントがパスワード π を知っている場合にのみ計算することができたものである。

【0040】本発明のもう1つの実施例を図5に示す。これも、サーバがパスワード π を所有せず、代わりに、パスワードベリファイアVを記憶するようなプロトコルを示す。ステップ501～514は、図4に関して前述したステップ401～414と同様である。ステップ516で、サーバは、 Z_q からランダムなcを選び、aを次のように計算する。

【数1】

$$c \in_R \{0, 1\}^k, a = g^{H'_0(A, B, c)} \bmod p$$

ただし、 H'_0 は、 $|q|+se$ cビットを出力しなければならないあるランダムハッシュ関数である。ステップ518で、サーバは、次式を計算する。

【数2】

$$k = c \oplus H_{2a}(A, B, m, \mu, a, \sigma, V^{H'_0(A, B, c)} \bmod p, V)$$

ただし、

【数3】

$$\oplus$$

$$c = k \oplus H_{2a}(A, B, m, \mu, a, \sigma, a^v, V)$$

ステップ526で、クライアントは、

【数5】

$$g^{H'_0(A, B, c)} \bmod p$$

を計算し、計算した値が、ステップ520でサーバから受信したaの値と同一であるかどうかをテストする。同一でない場合、サーバはクライアントに正しく認証されておらず、クライアントはプロトコルを終了する。計算されたaがステップ520でサーバから受信したaの値と同一である場合、クライアントは、サーバが真正なものであると判断し、プロトコルのステップ528に進み、 $k' = H_{2b}(A, B, m, \mu, \sigma, a, k, c, V)$ を計算する。ステップ530で、クライアントは、セッション鍵 $K=H_3(A, B, m, \mu, \sigma, c, V)$ を計算する。ステップ532で、クライアントは、 k' をサーバへ送信する。ステップ534で、サーバは、 $H_{2b}(A, B, m, \mu, \sigma, a, k, c, V)$ を計算し、計算した値が、ステップ532でクライアントから受信した k' の値と同一であるかどうかを判断する。同一でない場合、クライアントはサーバに認証されておらず、サーバはプロトコルを終了する。計算された k' が、ステップ532でクライアントから受信した k' の値と同一である場合、サーバは、クライアントが真正な

は排他的論理和(XOR)を表す。ステップ520で、サーバは、 μ, a, k をクライアントへ送信する。 μ, a, k を受信すると、ステップ522で、クライアントは、 $\sigma = \mu^x \bmod p$ を計算する。この時点で、クライアントおよびサーバはいずれも、共有秘密 σ を所有する。ステップ524で、クライアントは、次式を計算する。

【数4】

ものであると判断し、プロトコルのステップ536に進み、セッション鍵 $K=H_3(A, B, m, \mu, \sigma, c, V)$ を計算する。

【0041】直観的には、クライアントとサーバの認証はエルガマル暗号に基づいている。ここではエルガマル暗号については簡単に説明するとどめるが、さらに詳細には、T. ElGamal, "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms", IEEE Transactions on Information Theory, IT-31.4, pp. 469-472, 1985、に記載されている。一般に、メッセージMは、エルガマル暗号により $E(M) = (g^r, y^r M)$ と暗号化される。ただし、rはランダムな値であり、yは公開鍵であり、xを秘密鍵として、 $y = g^x$ である。暗号(A, B)は、 $D(A, B) = B/A^x$ として復号される。値を代入すると、 $B/A^x = y^r M / g^{rx} = g^{rx} M / g^{rx} = M$ となる。このように、エルガマル方式を用いて暗号化されたメッセージを復号するためには、秘密鍵が必要である。エルガマル暗号の第2版では、メッセージMは、次式のように暗号化される。

【数6】

$$E(M) = (g^r, H(y^r) \oplus M)$$

ただし、rはランダムな値であり、yは公開鍵であり、

x を秘密鍵として、 $y = g^x$ である。暗号 (A, B) は、この第2方式によれば、次式のように復号される。

【数7】

$$D(A, B) = H(A^x) \oplus B$$

値を代入すると、次式を得る。

【数8】

$H(A^x) \oplus H(y^r) \oplus M = H(y^r) \oplus H(y^r) \oplus M = M$
こうして、この場合も、エルガマル第2方式を用いて暗号化されたメッセージを復号するためには、秘密鍵が必要である。

【0042】ここで、「自己証明エルガマル暗号」を定義する。この暗号方式では、エルガマルの第2版を用いるが、 r は、単なる純粋にランダムな値の代わりに、 $r = H'(M)$ 、すなわち、復号される値のランダムハッシュとする。この場合、当事者が、暗号 (A, B) を受信し、それを復号して M を得ると、その当事者は、次式が成り立つかどうかをテストすることができる。

【数9】

$$A = g^{H'(M)}$$

注意すべき点であるが、任意の (A, B) は何かの暗号を定義するが、自己証明は、何がまさに暗号化されたかを、したがって、どの暗号化鍵が用いられているかを、暗号化をした者が知っていることを証明するものである。

【0043】ここで、図5に示したプロトコルを参照すると、当業者には認識されるように、ステップ516および518は、自己証明エルガマル暗号化を実行している。暗号化されているメッセージはランダム値 c であり、公開鍵はパスワードベリファイア V であり、秘密鍵は v である。暗号値はパラメータ k に代入されている。ステップ524で、クライアントは、サーバから受信した、暗号化された k を復号して c を抽出した後、ステップ526のテストを実行する。このテストが真である場合、サーバが正しいパスワードベリファイア V を所有していることが証明される。ステップ528で、クライアントは、 c の計算値を用いて k' を生成し、ステップ532で k' をサーバへ送信する。ステップ534でのサーバによるテストが真である場合、クライアントが正しい v を所有していることが証明される。秘密鍵 v がなければ、クライアントは復号を実行して、 k を計算する際に用いられるハッシュ関数への入力である c を得ることができないからである。

【0044】本発明のさらに別の実施例では、図2に示したプロトコルは、サーバがパスワード π を記憶せず、代わりに、値 $(H_1(A, B, \pi))$ r を記憶するように修正することが可能である。これは、追加の計算を必要とせず、2つのサーバ B_1 および B_2 で π を使用する素朴なユーザは、 B_1 に欠陥が生じて B_2 では自明に

は脆弱にならないという利点がある。

【0045】本発明の発明者は、本発明による相互認証および鍵交換プロトコルが安全であることを証明した。証明の直観的な概略は次の通りである。直観的には、次のことを証明しなければならない。

【0046】(1) パスワードを共有しプロトコルに従う二当事者が互いを認証し長い共有秘密を得る。

【0047】(2) Diffie-Hellmanプロトコルが安全であることを仮定すると、本発明のプロトコルは、信頼される当事者との「理想世界プロトコル」と同程度に安全である。「理想世界プロトコル」では、正当な二当事者は、互いにコネクションを設定し、信頼される当事者に、このコネクションを安全にするための長い共有秘密を生成してもらうことができるが、敵は、設定されたコネクションごとに1回（安全にされる前に）共有パスワードを信頼される当事者に問い合わせることが可能である。（直観的には、これは、敵が、パスワードのランダム推測を行い、自分自身を認証してもらおうとすることをモデル化している。）

【0048】(1) の部分は、プロトコルを見れば明らかである。

【0049】(2) の部分は、より難しい。パスワードを知らずに、理想世界の信頼される当事者を用いだけで、実プロトコルをシミュレートし、このシミュレーションを攻撃する敵は、実世界の実プロトコルを攻撃する敵とは区別不能であるようにすることができることを示す。（これは、“multi-party simulatability”（複数当事者シミュレート可能性）技法という、よく知られた暗号理論の証明技法であり、D. Beaver, “Secure Multiparty Protocols and Zero-Knowledge Proof Systems Tolerating a Faulty Minority”, Journal of Cryptology, 4(2), pp. 75-122, 1991, に記載されている。）

【0050】技術的には、われわれのモデルは、すべてのハッシュ関数が完全にランダムであり、その関数を使用するときにはいつも、シミュレータは入力を調べ、出力をセットすること（これらの出力がランダムにセットされる限り）を仮定している。

【0051】われわれのシミュレータの一般的な考え方は、単に、互いに通信する正当な二当事者間の長い共有秘密を模造し、他のすべての会話で、パスワードに関する推測を（敵のハッシュ関数問合せを検査することにより）検出し、それらの推測を、信頼される当事者への「テストパスワード」問合せに変えようとするものである。これの難しい部分は、敵が、設定されたコネクションあたり複数回のパスワード推測を行うことができないことを示すことである。これを示すためには、敵にそれが可能であるとする、Diffie-Hellman問題が解けてしまう（具体的には、（未知の x および y に対して） $X = g^x$ 、および $Y = g^y$ となる値 $X, Y, Z \in G$ をとり、 $Z = g^{xy}$ であるかどうかを判定することができる）こ

とを示す。

【００５２】上記の詳細な説明は、単なる例示であつて、限定的なものではないと理解されるべきである。ここで説明した実施例は、本発明の原理の単なる例示であり、当業者であれば、本発明の技術的範囲および技術思想から離れることなく、さまざまな変形例を考えることが可能である。例えば、上記のプロトコルにおいて、いくつかのパラメータがハッシュ関数を評価する際に用いられている。注意すべき点であるが、発見的セキュリティにはそのすべてのパラメータが必要とされるわけではなく、付加的パラメータは、プロトコルが形式的に安全であることを証明することを可能にするためのものである。例えば、図２のプロトコルのステップ２１６、２２２、２２４、２３０、２２６および２３２、図３のプロトコルのステップ３１６および３２４、図４のプロトコルのステップ４１６、４２２、４３８および４４０、ならびに図５のプロトコルのステップ５３０および５３６は、プロトコルが発見的に安全であるためには、ハッシュ関数においてパラメータ σ しか必要でないと考えられる。しかし、上記の付加的パラメータの使用により、プロトコルは形式的に安全であることが証明される。同様に、図４のプロトコルのステップ４２８および４３４は、プロトコルが発見的に安全であるためには、ハッシュ関数においてパラメータ σ および a しか必要でないと考えられる。しかし、上記の付加的パラメータの使用により、プロトコルは形式的に安全であることが証明される。同様に、図５のプロトコルのステップ５１８および５２４は、プロトコルが発見的に安全であるためには、パラメータ σ 、 a 、および

【数１０】

$$V^{H'_0}(A, B, c)$$

しか必要でないと考えられる。しかし、上記の付加的パラメータの使用により、プロトコルは形式的に安全であることが証明される。最後に、図５のプロトコルのステップ５２８および５３４は、プロトコルが発見的に安全であるためには、ハッシュ関数においてパラメータ σ および c しか必要でないと考えられる。しかし、この場合も、上記の付加的パラメータの使用により、プロトコルは形式的に安全であることが証明される。

【００５３】

【発明の効果】以上述べたごとく、本発明によれば、安全であることが証明可能な、安全な、パスワードのみによる相互ネットワーク認証プロトコルが実現される。

【図面の簡単な説明】

【図１】従来のDiffie-Hellman鍵交換プロトコルを示す図である。

【図２】両当事者が共有パスワードを所有する、本発明の明示的認証実施例による通信プロトコルを示す図である。

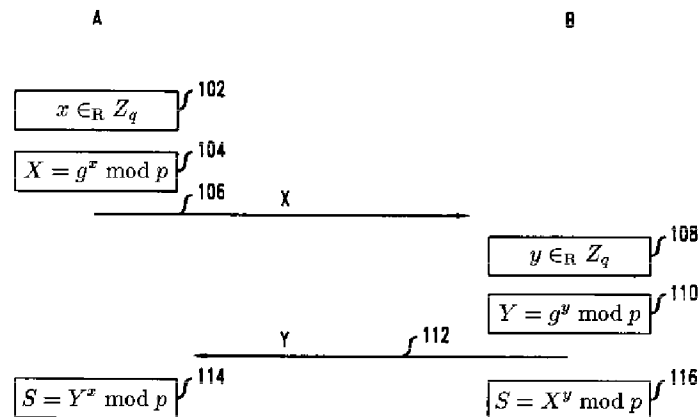
【図３】両当事者が共有パスワードを所有する、本発明の暗黙的認証実施例による通信プロトコルを示す図である。

【図４】一方当事者がパスワードを所有し他方当事者がパスワードベリファイアを所有する、本発明の明示的認証実施例による通信プロトコルを示す図である。

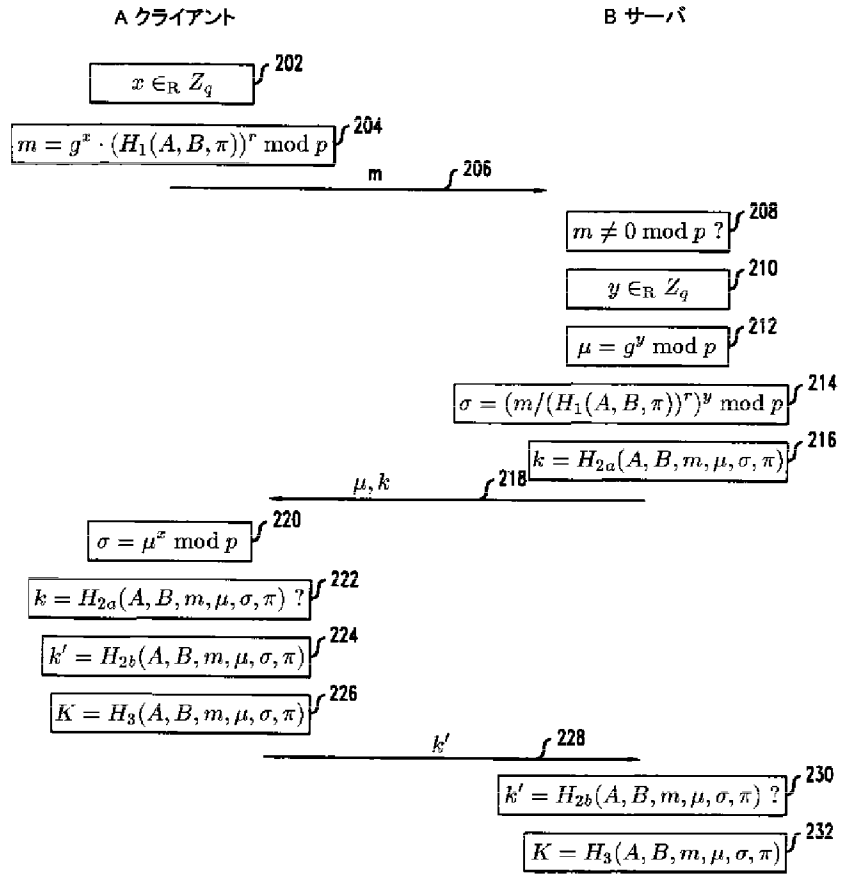
【図５】一方当事者がパスワードを所有し他方当事者がパスワードベリファイアを所有する、本発明のもう１つの明示的認証実施例による通信プロトコルを示す図である。

【図１】

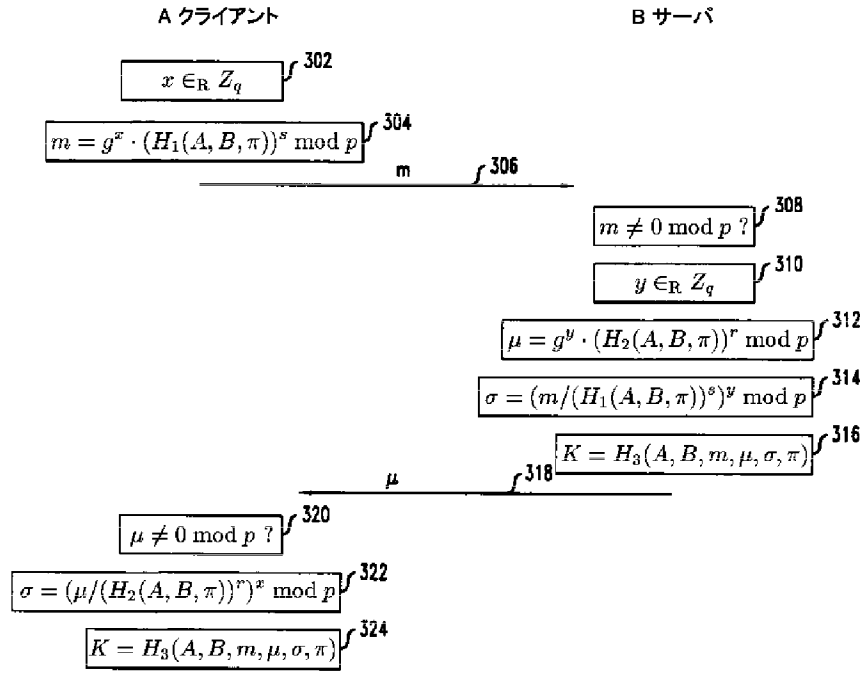
(従来技術)



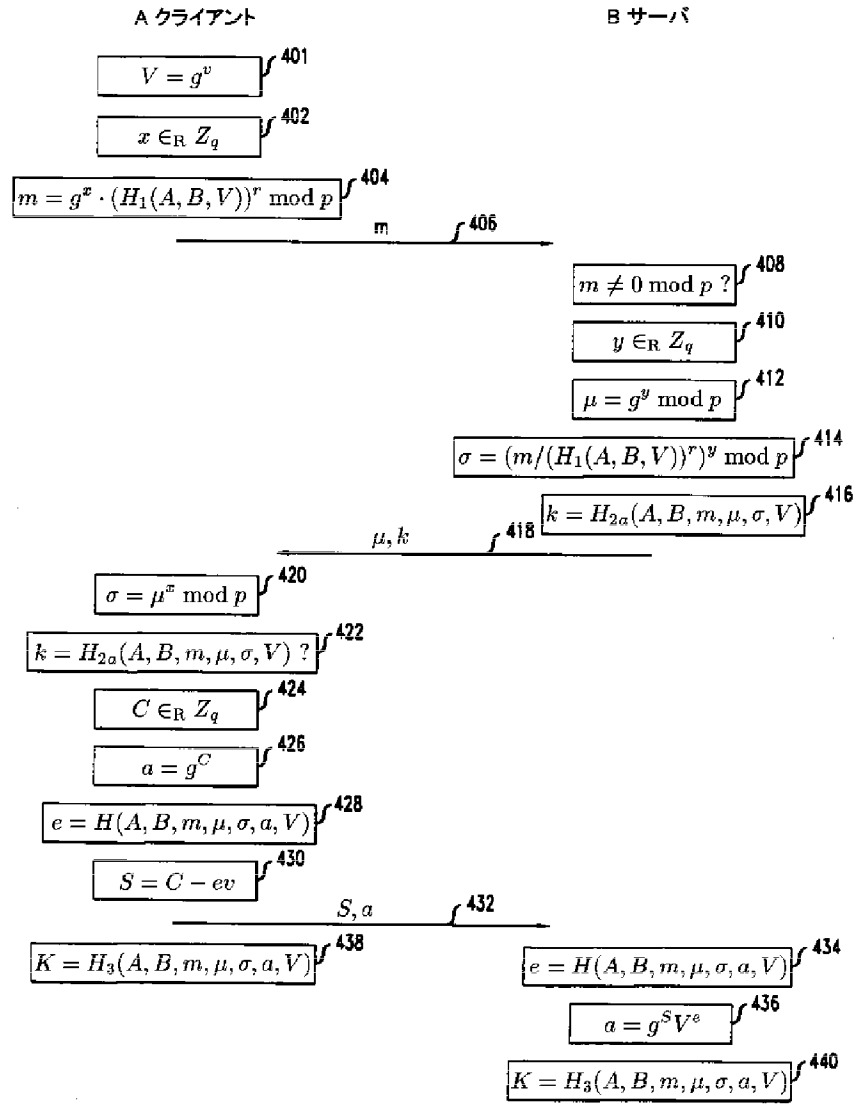
【図2】



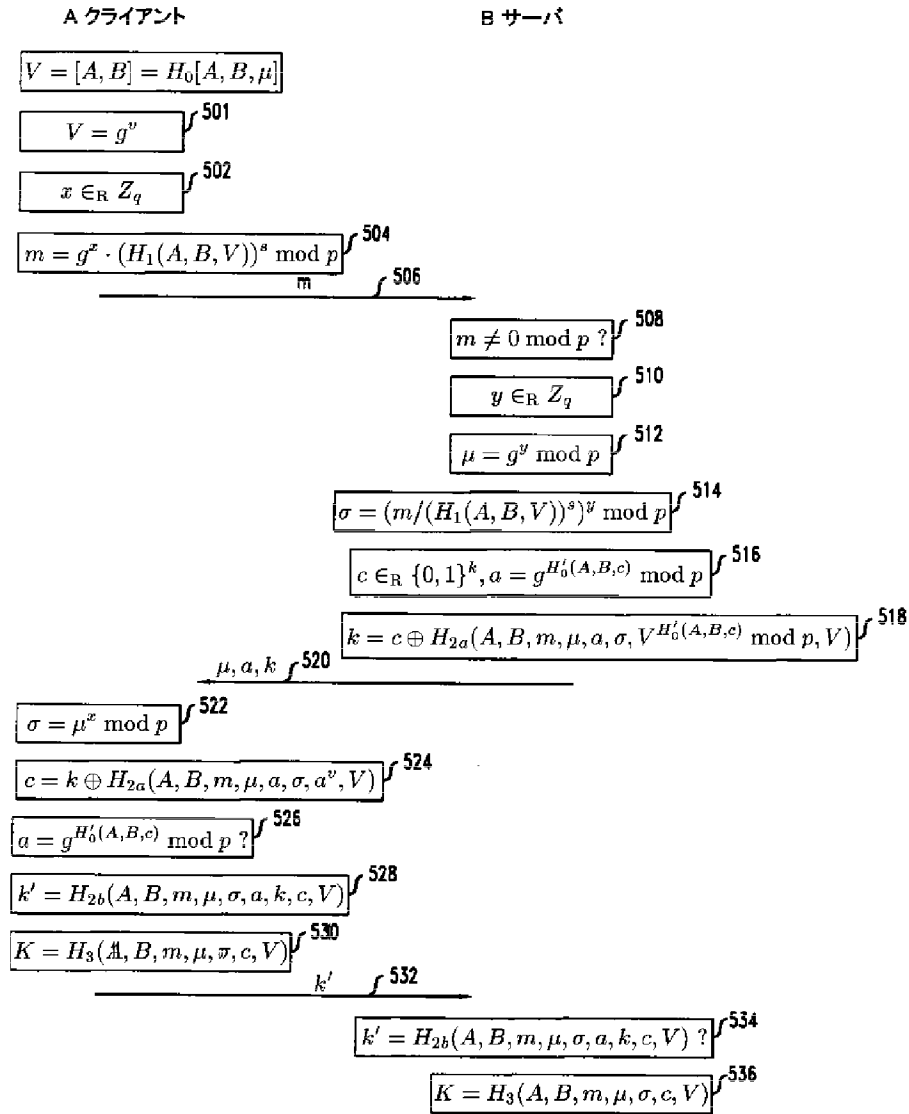
【図3】



【図4】



【図5】



フロントページの続き

(71)出願人 596077259
600 Mountain Avenue,
Murray Hill, New Je
rsey 07974-0636 U. S. A.
(72)発明者 エリック グロッセ
アメリカ合衆国、07922、ニュージャージ
ー州、パークリー ハイ츠、ノース ロー
ド 140

(72)発明者 ビクター ウラジミール ボイコ
アメリカ合衆国、10952、ニューヨーク州、
マンジー、ルート306、244
(72)発明者 フィリップ ディー、マッケンジー
アメリカ合衆国、07040 ニュージャージ
ー州、メイプルウッド、カールトン コー
ト 11
(72)発明者 サーバー パテル
アメリカ合衆国、07045 ニュージャージ
ー州、モントビル、ミラー レーン 34